PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-044630
(43)Date of publication of application : 16.02.1996

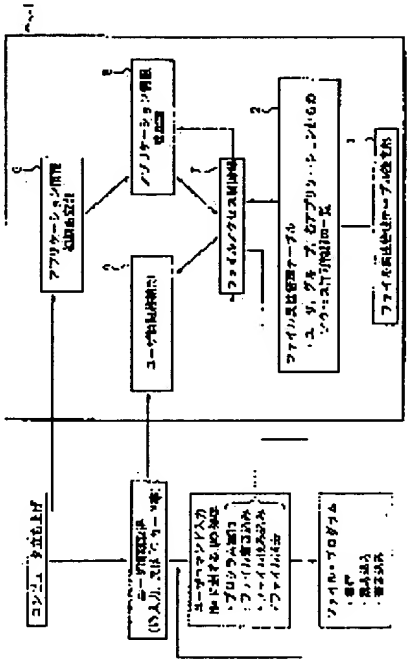
(51)Int.Cl. G06F 12/14
G06F 12/00

(21)Application number : 06-182575
(22)Date of filing : 03.08.1994

(71)Applicant : NRI & NCC CO LTD
(72)Inventor : MASHITA TATSUMI
ONO KIYOSHI

(54) DEVICE FOR CONTROLLING FILE ACCESS AND METHOD THEREOF

(57)Abstract:
PURPOSE: To provide a device for controlling file access and method thereof with which data can be effectively prevented from being destroyed over a wide range by the malfunction of a software composed of plural application softwares or the execution of any troubled program.
CONSTITUTION: This device is provided with a file attribute managing table 2 defining a user to perform each file, application software and the kind of processing to that file, user information storage means 4 for storing the information of the user under executing the software, application information managing means 5 for managing the information of the application software under being executed, and file access control means 7 for permitting or inhibiting the instruction of access to the file while referring to the user information storage means 4, application information managing means 5 and file attribute managing table 2.



LEGAL STATUS

[Date of request for examination] 23.03.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

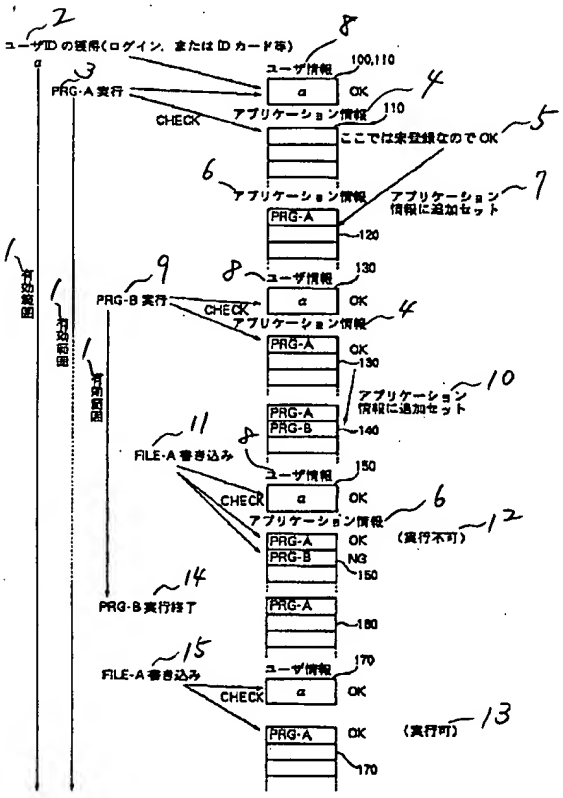
[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

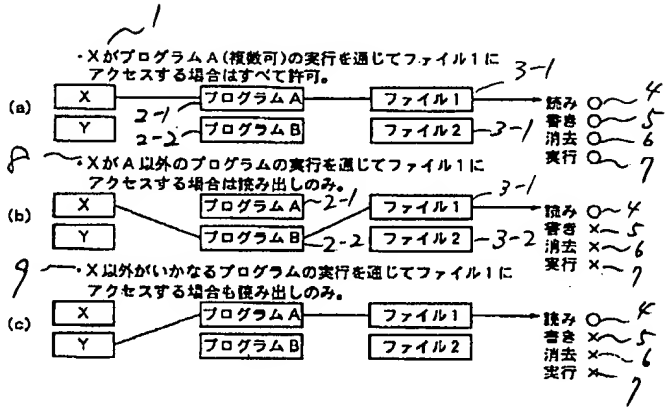
[Date of extinction of right]

ファイルアクセス制御装置およびその制御方法

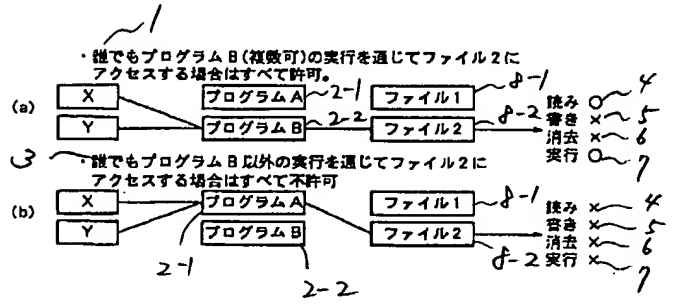
【図３】



【図４】



【図５】



ファイルにアクセス。この結果、アプリケーションソフトウェアの誤作動や不具合があるプログラムやウィルス感染による広範なデータ破壊を実質的に防止することができる。

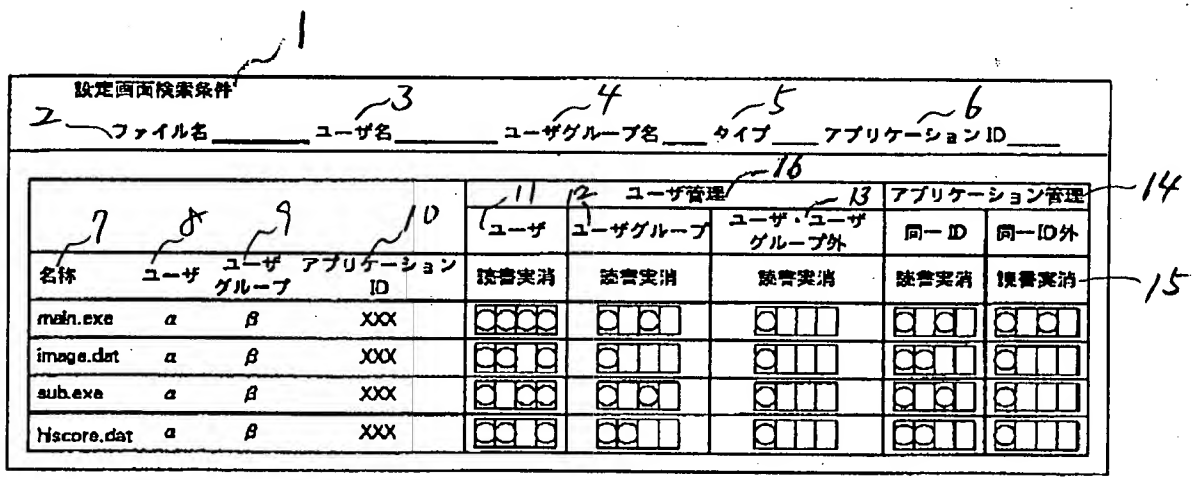
【００７３】

【発明の効果】上記説明から明らかなように、本願請求項１および請求項３に係るファイルアクセス制御装置および制御方法は、ファイル属性管理テーブルと、アプリ

【符号の説明】

1　ファイルアクセス制御装置
2　ファイル属性管理テーブル
3　ファイル属性管理テーブル設定部
4　ユーザ情報格納部
5　アプリケーション情報管理部
6　アプリケーション情報初期設定部
7　ファイルアクセス制御部

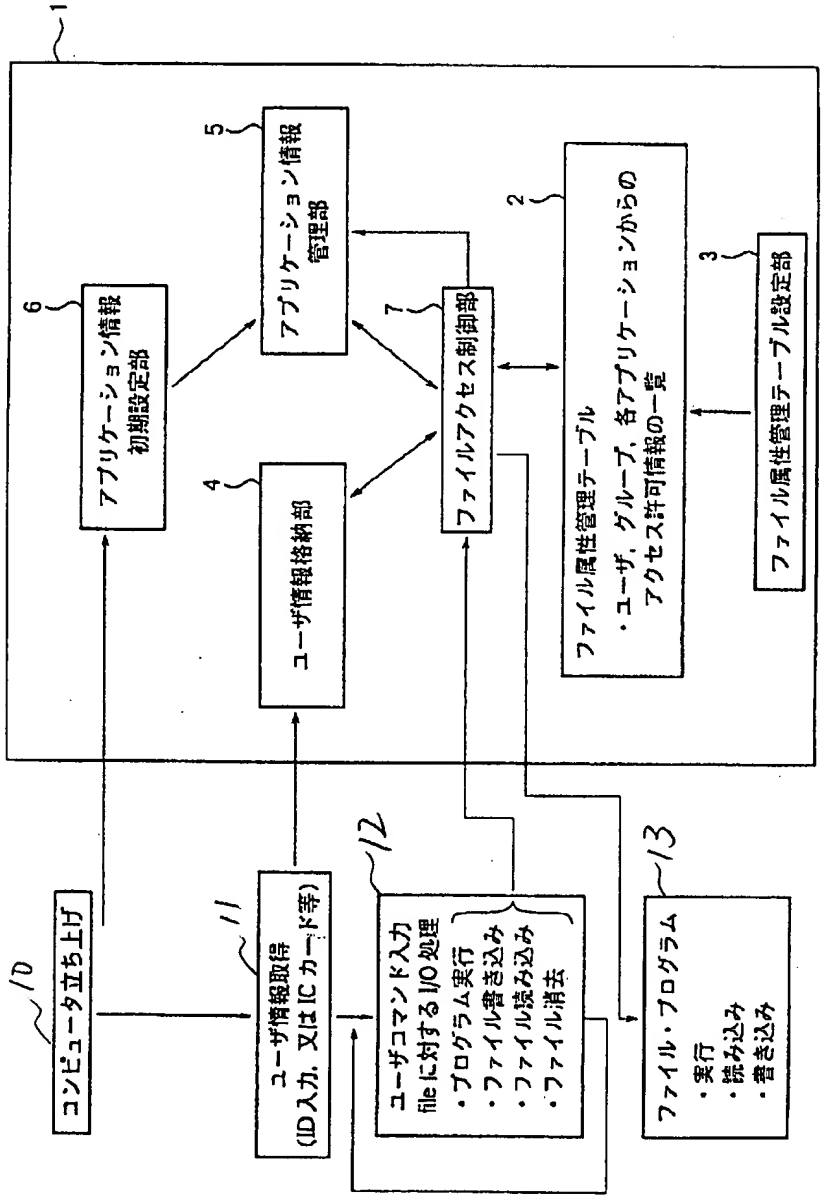
【図２】



ファイルアクセス制御装置およびその制御方法

特開平8－44630

【図1】



Japanese Patent Laid-open No. 8-44630

[Title of the Invention]

DEVICE FOR CONTROLLING FILE ACCESS AND METHOD THEREOF

[Abstract]

[Purpose]

To provide a file access control device capable of effectively preventing a wide range of data destruction which otherwise would occur when some of a plurality of pieces of application software malfunction or an imperfect program is executed, and a control method thereof.

[Constitution]

There are provided a file attribute management table 2 where each file is defined as to the users and application softwares which are permitted to access the file and the types of processing permitted; user information storage means 4 which stores information about the user which is executing the software; application information management means 5 for managing information about the application software which is being executed; and file access control means 7 which permits or prohibits access instructions by referring to the user information storage means 4, the application information management means 5 and the file attribute management table 2.

[What is Claimed is]

[Claim 1]

A file access control device, comprising:

a file attribute management table wherein each file constituting software is defined as to users and application software permitted to access the file, and the types of processing permitted on the file;

user information storage means for storing information about the user who is executing the application software;

application information management means for managing information about the application software which is being executed;

when an instruction to access the file is issued, file access control means for referring to the user information storage means and the application information management means in order to identify the user and application software that have issued the instruction; and enabling or disabling the instruction in accordance with the definitions given in the file attribute management table.

[Claim 2]

A file access control device according to claim 1, further comprising a file attribute management table setting means by which each file can be defined through a computer screen as to the users and application software

permitted to read from, write to, execute and/or delete the file.

[Claim 3]

A file access control method, comprising the steps of:

defining a user and application software permitted to read from, write to, execute and/or delete files constituting a software on an each file basis and storing the definitions into computer's storage means;

having a user enter user information when the user executes the software and registering the user information as the current user executing the software;

monitoring respective accesses to files and registering the current application software being executed; and

when an instruction to access the file is issued, identifying the user and application software that have issued the access instruction; and enabling or disabling the access instruction in accordance with the definitions given to the file to be accessed by the user and application software.

[Claim 4]

A file access control method according to claim 3, further comprising the step of defining a file of the software through a computer screen as to the users and application softwares permitted to read from, write to,

execute and/or delete the file according to the situation of the software while the software is used.

[Detailed Description of the Invention]

[0001]

[Field of the Invention]

The present invention relates to a device and method for controlling accesses to program and data files in order to prevent malfunctioning programs and illegally executed programs from damaging programs and data, and viruses from causing large-scale data destruction by infection, and more particularly to such a file access control device and method that each file containing a program or data is defined with regard to the users and programs (hereinafter referred to as application software or simply as applications) permitted to access the file in order to prevent a chain of executed applications from causing large-scale data destruction.

[0002]

[Prior Art]

Generally, software internally contains a plurality of program files and data files and is designed so that its execution advances while execution is switched among these programs (applications).

[0003]

One example of such software is game software. Game

software is generally designed so that the initial screen successively evolves to new screens which respectively allow the user to perform specific operations and as a whole to enjoy the software as a game. Not limited to game softwares, this technique is now used in various types of software.

[0004]

This type of software is configured in such a manner that applications are mutually and complicatedly related with other applications. For example, one application can be read out or executed by a plurality of other applications and a program/data file can be updated or deleted by another application.

[0005]

However, such software has a risk that a misoperation or execution of an application having an imperfect portion may damage an unrelated program/data file (which should not be accessed by the application), and further may successively damage a chain of unrelated files, resulting in serious data destruction throughout the software.

[0006]

In Particular, the problem of virus infection has become an issue these days. A virus intentionally tries to cause the above-mentioned phenomenon. It may cause a wide-range chain of data destruction by having a hidden data

destruction program executed.

[0007]

To cope with this, devices and methods have been proposed to control applications' accesses to program/data files. A conventional file access control device/method controls accesses to each file by such a scheme that write-permitted programs/data and write-prohibited programs/data are respectively loaded into access-permitted and access-prohibited memory regions and the originator of each access instruction and each program/data to be accessed is managed by their addresses in the memory (See Japanese Patent Laid-open No. Sho 57-71597).

[0008]

On the other hand, there has also been an access control device/method which enables or disables an access to program/data files depending on the user. This is intended to protect software by giving the access right to the user only when the user is authorized to use the file. Practically, an access to a file is enabled or disabled by checking the user's identifier either on an each user basis or on an each user group basis.

[0009]

[Problems to be Solved by the Invention]

However, in the above-mentioned conventional method where write-permitted programs/data and write-prohibited programs/data are respectively loaded into access-permitted

and access-prohibited memory regions and an access to a file is managed according to the file's address in the memory, many restrictions are imposed on the configuration of the software. Complex software requires detailed configuration. For example, some program/data file must be protected from write accesses by specific programs but not protected from write accesses by the other programs. Such detailed settings cannot be done by the above-mentioned conventional method since determining whether to permit or prohibit write to a file depends on the location of the file in the memory.

[0010]

In addition, once write permission or write prohibition is set, it is not possible to flexibly change the setting according to the execution state of the software. Further, it is not possible to prohibit specific users from accessing a file or permit specific users to access the file.

[0011]

On the other hand, in the above-mentioned user check method which determines whether to enable or disable an application to access a file by checking the user's identifier information, the user can access any file of the software once the user is identified as an authorized user. Therefore, if an authorized user issues a wrong execution instruction by a misoperation or runs a program containing

an imperfect portion, a chain of data destruction is sometimes caused by applications.

[0012]

In particular, if software is infected with a virus, the above-mentioned conventional method cannot prevent the virus from causing wide-range data destruction since the software hides the imperfect program and allows authorized users to execute the program.

[0013]

Therefore, it is an object of the present invention to overcome the deficiencies of the above-mentioned conventional techniques and provide a file access control device and a control method thereof capable of effectively preventing a software composed of a plurality of pieces of application software from having wide-range data destruction caused either by a misoperation or by execution of an imperfect program.

[0014]

[Means for Solving the Problems]

To achieve the above-mentioned object, a file access control device according to claim 1 in the present specification is characterized in that it comprises: a file attribute management table wherein each file constituting software is defined as to users and application software permitted to access the file, and the types of processing permitted on the file; user information storage means for

storing information about the user who is executing the application software; application information management means for managing information about the application software which is being executed; when an instruction to access the file is issued, file access control means for referring to the user information storage means and the application information management means in order to identify the user and application software that have issued the instruction; and enabling or disabling the instruction in accordance with the definitions given in the file attribute management table.

[0015]

A file access control device according to claim 2 in the present specification is characterized in that it is a file access control device according to claim 1, mentioned above, and further comprises a file attribute management table setting means by which each file can be defined through a computer screen as to the users and application software permitted to read from, write to, execute and/or delete the file.

[0016]

A file access control method according to claim 3 in the present specification is characterized in that it comprises the steps of: defining users and application software permitted to read from, write to, execute and delete files constituting a software on an each file basis

and storing the definitions into computer's storage means; having a user enter the user information when the user executes the software and registering the user information as the one of the current user executing the software; monitoring respective accesses to files and registering the current application software being executed; and when an instruction to access the fileis issued, identifying the user and application software that have issued the access instruction; and enabling or disabling the access instruction in accordance with the definitions given to the file to be accessed by the user and application software.

[0017]

A file access control method according to claim 4 in the present specification is characterized in that it is a file access control method as claimed in claim 3 and further comprises the step of defining a file of the software through a computer screen as to the users and application software permitted to read from, write to, execute and/or delete the file according to the situation of the software while the software is used.

[0018]

[Operation]

A file access control device and method as claimed in claims 1 and 3 in the present specification defines/registers each file of the software in advance to a file attribute management table as to the users and

applications permitted to access the file and the types of processing on the file, stores information about the current user executing the software into user information storage means and stores information about the current application software being executed into application information management means. When an instruction is issued to access a file, it is possible to identify, by the user information storage means and application information management means, the user and application software that have issued the access instruction and enabling or disabling the access instruction in accordance with the file attribute management table.

[0019]

Therefore, the above-mentioned file access control device and method according to the present invention can determine the users and applications permitted to access each file of the software and the types of processing permitted on the file (for example, read, write, execute, delete, etc.), and therefore can guarantee that processing is performed by proper operations of the applications constituting the software and prevent data destruction in a chain of files from being brought about by a malfunction or execution of an imperfect program.

[0020]

In addition, a file access control device and method according to claims 2 and 4 in the present specification

allows the condition for accessing any file to be changed as necessary according to the situation of the software while the software is used since each file constituting the software respectively can be re-defined anytime through a computer screen as to a user and an application software permitted to read from, write to, execute and delete the file. Further, due to the flexibility allowing file access conditions to be defined arbitrarily, it is possible to provide a file access control device and method applicable to various types of software.

[0021]

[Embodiments]

In the following, embodiments of a file access control device and a control method thereof according to the present invention will be explained with reference to the attached drawings.

[0022]

FIG. 1 shows an embodiment of a file access control device according to the present invention. Its configuration and processing flows are indicated.

[0023]

Although a file access control device 1 in this embodiment can be configured by using another hardware instead of the computer to execute each software, it is preferably configured by using the processor, storage, input output device or the like of the computer in which

software is executed.

[0024]

As indicated in FIG. 1, the file access control device 1 comprises: a file attribute management table 2 where each file of a software to be executed is defined as to the users, user groups and application softwares permitted to access the file; a file attribute management table setting unit 3 to set the contents of the file attribute management table 2; a user information storage unit 4 to store information about the current user executing the software; an application information management unit 5 to recognize the application software which has issued the current command being executed; an application information initialization unit 6 to initialize the application information management unit 5; and a file access control unit 7 to enable or disable an instruction issued to access a file by referring to the user information storage unit 4, the application information management unit 5 and the file attribute management table 2.

[0025]

Then let us explain operations of the file access control device 1 configured as described above while software is executed.

[0026]

Assume that the file attribute management table 2 is

already set up. When the computer is started, the application information initialization unit 6 receives the start signal and initializes the application information. Usually, the application information management unit 5, if initialized, has no application software registered therein.

[0027]

After the computer is started, the user is immediately asked to enter user information. This intends to preliminarily reject the illegal users before the software is executed.

[0028]

User information can be acquired by such known methods as asking the user to enter his user ID or insert his IC card for collation.

[0029]

If the acquired user information is information of an authorized user, the user is permitted to use the software and the acquired user information is stored in the user information storage unit 4.

[0030]

Run by the user permitted to use the software as above, various types of processing are performed including processing for the input/output of a great number of files (program files and data files) constituting the software.

[0031]

In the file access control device 1 of this

embodiment, any file I/O instruction is once put into the file access control unit 7 before the file is accessed, in order to determine whether to enable or disable the access in advance. The supported types of I/O processing are execution of a program, write to a file, read from a file and delete in a file.

[0032]

If the file access control unit 7 receives a file I/O processing instruction to read in a new program for execution, it registers the new program name to the application information management unit 5. By this, the latest information can always be maintained about the current application being executed.

[0033]

If a file I/O processing instruction is received which demands to change or delete contents of a specific file, the file access control unit 7 refers to the user information storage unit 4, the application information management unit 5 and the file attribute management table 2 in order to enable or disable the instruction by checking whether the instruction is originated from a user and application software that are permitted to access the file and whether the type of processing is permitted on the file.

[0034]

Then, let us explain the contents of the file attribute management table used as the basis for the above-

mentioned judgement.

[0035]

FIG. 2 illustrate a screen provided by the file attribute management table setting unit 3 for setting the contents of the file attribute management table 2.

[0036]

The upper part of the screen in FIG. 2 is used to enter search conditions to retrieve a setting screen. If at least one of a filename, a user name, a user group name, a type and a file ID is entered, the setting screens of files which match the search conditions are displayed out of the file attribute management table by the file attribute management table setting unit 3.

[0037]

Below the above-mentioned part to enter search conditions, there is a part to set contents of the file attribute management table.

[0038]

The left area of the part for setting contents of the file attribute management table is provided to specify the names of the files to be accessed (main.exe, image.dat, sub.exe. and hiscore.dat in this example), users (α) and user groups (β) permitted to access the respective files and IDs (XXXX) given to the respective files as applications.

[0039]

On the right side of the above-mentioned area for designating files, users, etc., there is an area for setting access conditions to each file. This access condition setting area is largely divided into two sections. One is for user management while the other is for application management.

[0040]

The user management setting section is further divided into three columns which are respectively for users, user groups and others. Note that this division is made for an example. Further detailed division is possible. For example, the section may be divided further so that access conditions can be set for each user.

[0041]

In each of the divided user management columns, there is shown conditions to be permitted or prohibited for respective types (read, write, execute and/or delete) of processing on the file can be set.

[0042]

Likewise, the application management setting section is divided into two columns which are respectively for the application having the specified ID and other applications so that the conditions to be permitted or prohibited for respective types of processing can respectively be set for accesses from that application and accesses from other applications. Further detailed division is also possible

here. For example, the section may be divided further so that access conditions can be set for each application ID.

[0043]

The setting example in FIG. 2 allows the authorized user α and user group β to execute the file "main.exe" while allowing the other users and user groups only to read the file "main.exe". The user α is allowed to change and delete contents of the file "main.exe", too. It is also programmed that all application software are allowed to start and execute the file "main.exe".

[0044]

To the contrary, the right to read from and write to the file "image.dat" is given only to the application having the specified application ID. The user α is the only one who can write to the file.

[0045]

In the case of the file "sub.exe", the user α and the members of the user group β are allowed to execute the file "sub.exe" although it is always necessary to use the file given the application ID "main.exe" (same as the specified ID) when executing the file "sub.exe". The user α is allowed to delete contents of the file "sub.exe" but not allowed to write to the file in order to prevent misoperation.

[0046]

Finally, read from and write to the file

"hiscore.dat" can be done by the user α and the members of the user group β by using either file "main.exe" or file "sub.exe". The right to delete contents of the file "hiscore.dat" is given only to the user α.

[0047]

As described above, by setting the user management and application management sections and the types of processing to be permitted of the file attribute management table 2 according to the present invention, it is possible to freely specify what users and applications are to be permitted to access respective files and what types of processing are to be permitted on these files. In addition, since the file attribute management table 2 can be set when necessary by using the file attribute management table setting unit 3, it is possible, for example, to flexibly add/change users and applications permitted to access a file when the design of the software is changed. Further, it is possible to freely specify conditions to access that file. Thus, a file access control device and method according to the present invention can have enough wide usability for application to various types of softwares.

[0048]

Then, let us explain the access control by the file access control unit 7 by following practical operations of an application.

[0049]

FIG. 3 illustrates a control sequence done by the file access control unit 7. Assume the software shown in FIG. 3 is composed of a plurality of pieces of application software configured so as to be activated and executed successively in some order by user commands.

[0050]

In FIG. 3, a main application PRG-A is activated and a file FILE-A is updated by this main application PRG-A.

[0051]

This case assumes that it is specified by the file attribute management table 2 that the user α can directly activate the main application PRG-A, activate the sub application PRG-B through the main application PRG-A and update the data file FILE-A through the sub application PRG-B.

[0052]

In FIG. 3, if the user ID α is acquired after the computer is started, it is checked if the user ID α is the ID of an authorized user. If so, the ID is stored in the user information storage unit 4 (Step 100).

[0053]

The stored user ID α remains effective until a new user ID is acquired, that is, unless you terminate the software and restart the computer.

[0054]

Note that as explained in FIG. 1, starting the

computer resets the application information management unit 5 to the initial state in which no application is registered although this is not indicated in FIG. 3.

[0055]

Then, if a command is entered which demands to execute the main application PRG-A, the file access control unit 7 detects information about this access to the PRG-A and checks the user information management unit 4 and the application information management unit 5 to detect the user and application that are trying to execute the PRG-A (Step 110).

[0056]

In this case, since no application has been registered to the application information management unit 5 while the user information α is acquired from the user information storage unit 4, the instruction is found a direct instruction by the user α to activate the PRG-A. As mentioned above, it is defined in the file attribute management table 2 that the user α is permitted to directly activate the PRG-A. Thus, the main application PRG-A is executed and at the same time the PRG-A is registered to the application information management unit 5 as the current application being executed (Step 120).

[0057]

Application information (PRG-A) registered to the application information management unit 5 remains effective

in the application information management unit 5 until execution of the PRG-A is completed.

[0058]

Then, during execution of the software, assume that the PRG-A issues an instruction demanding to execute the sub application PRG-B. The file access control unit 7 detects this instruction involving an access to the sub application PRG-B, and asks the user information storage unit 4 and the application information management unit 5 about the user and application that have originated this instruction demanding to execute the sub application PRG-B (Step 130).

[0059]

As the result, the file access control unit 7 is notified that this PRG-G execution instruction is issued by the user α via the main application PRG-A, and enables the execution of the sub application PRG-B according to the conditions set in the file attribute management table 2.

[0060]

After the sub application PRG-B is put under execution, the file access control unit 7 additionally registers the PRG-B to the application information management unit 5 as the current application being executed (Step 140).

[0061]

Then, assume that the sub application PRG-B during

execution issues an instruction to rewrite the data file FILE-A. The file access control unit 7 asks the user information storage unit 4 and the application information management unit 5 about the user and application that have issued the write instruction. As the result, the file access control unit 7 determines that the instruction has been issued by the user α to rewrite contents of the data file FILE-A via the sub application PRG-B, and disables this write instruction according to the conditions set in the file attribute management table 2 (Step 150).

[0062]

Operations to be performed after the sub application PRG-B was executed will be described.

Upon completion of the sub application PRG-B, the file access control unit 7 detects the completion and deletes the registration of the sub application PRG-B in the application information management unit 5 (Step 160).

[0063]

In addition, completion of the sub application PRG-B resumes execution of the suspended main application PRG-A. Assume that the resumed main application PRG-A issues an instruction to write to the data file FILE-A. In this case, the file access control unit 7 determines that the user α is intending to rewrite the data file FILE-A via the main application PRG-A, and enables the write instruction according to the conditions set in the file attribute

management table 2 (Step 170).

[0064]

FIGS. 4 and 5 illustrate control types by a file access control device and method according to the present invention.

[0065]

Examples in FIG. 4 indicate that file accesses can be controlled mainly by user management in such a manner that only a specific user X can update/delete a file 1 only by operating a program A.

[0066]

FIG. 4 (a) indicates the types of file processing the user X can perform via the program A. The user X can perform any of read, write, delete and execution.

[0067]

To the contrary, FIG. (b) indicates the types of file processing the user X can perform via an unspecified program B. In this case, only read is permitted in order to protect data from destruction by malfunction of the program B.

[0068]

FIG. 4 (c) indicates the types of file processing users other than the user X can perform. In this case, only read is permitted even when the program A is used.

[0069]

FIG. 5 indicates application management-based file

access control in such a manner that only a specific program B is permitted to execute a file 2.

[0070]

In FIG. 5 (a), any of the users X and Y can read/execute the file 2 only via the program B. To the contrary, in FIG. 5 (b), any of the users X and Y cannot read/update/delete/execute the file 2 via the program A.

[0071]

As indicated above, user management-based file access control combined with application management-based file access control provides various modes of file access control and therefore makes it possible to define detailed relationships among applications constituting a software.

[0072]

Accordingly, although data destruction caused by, for example, an imperfect program executed mistakenly can not be eliminated, it is possible to isolate the influence to very limited files. As a result, it is substantially possible to prevent a wide range of data destruction from being caused by malfunctioning application softwares, imperfect programs, infecting viruses, etc.

[0073]

[Effects of the Invention]

As understood from the explanation provided so far, it is clear that a file access control device and method according to claims 1 and 3 in the present specification

can prevent a file from being updated or deleted by execution of an unrelated program (application) since a file attribute management table, application information management means, user information storage means and file access control means can operate to limit not only the users and applications accessible to the file but also the types of processing permitted on the file. Substantially, it is therefore possible to prevent wide and serious data destruction from being caused by a computer malfunction, imperfect programs or infecting viruses, etc.

[Brief Description of the Drawings]

FIG. 1 is a schematic diagram showing the configuration of a file access control device according to an embodiment of the present invention and flows of information among the component units of the device;

FIG. 2 shows an example of a file attribute management table setting screen according to the present invention;

FIG. 3 is a flowchart where operation flows of an file access control device according to the present invention is indicated using practical applications;

FIG. 4 indicates modes of access to a specific file implemented mainly by user management; and

FIG. 5 indicates modes of access to a specific file implemented mainly by application software management.

Description of the Symbols

1. File access control device
2. File attribute management table
3. File attribute management table setting unit
4. User information storage unit
5. Application information management unit
6. Application information initialization unit
7. File access control unit

FIG. 1

1. File access control device

2. File attribute management table

A list of access permission information from user, group and respective applications

3. File attribute management table setting unit

4. User information storage unit

5. Application information management unit

6. Application information initialization unit

7. File access control unit

10. Computer started

11. User information acquired (ID entry, IC card, etc.)

12. User command entered

I/O processing on a file

· Program execution

· File write

· File read

· File delete

13. File/program

· Execution

· Read

· Write

FIG. 2

1. Setting screen search condition
2. File name
3. User name
4. User group name
5. Type
6. Application ID
7. Name
8. User
9. User group
10. Application ID
11. User
12. User group
13. Unspecified users/user groups
14. Application management
15. R W E D
16. User management
17. Identical ID
18. Not identical ID

FIG. 3

1. Effective
2. User ID acquired (Login, ID card, etc.)
3. PRG-A execution
4. Application information

5. OK here due to no registration

6. Application information

7. Added to application information

8. User information

9. PRG-B execution

10. Added to application information

11. FILE-A write

12. (Write not permitted)

13. (Write permitted)

14. PRG-B execution end

15. FILE-A write

FIG. 4

1. User X's accesses to the file 1 via the program A (or two or more specific programs) are all permitted.

2-1. Program A

2-2. Program B

3-1. File 1

3-2. File 2

4. Read

5. Write

6. Delete

7. Execute

8. When user X accesses the file 1 via any program other than program A, only read is permitted.

9. When the file 1 is accessed by any other user than X,

only read is permitted, too.

FIG. 5

1. Anyone can access the file 2 via the program B (or two or more specific programs).

2-1. Program A

2-2. Program B

3. Anyone cannot access the file 2 via any program other than program B.

4. Read

5. Write

6. Delete

7. Execute

8-1. File 1

8-2. File 2